ASTRONOMICAL DATA ANALYSIS SOFTWARE AND SYSTEMS XIV
ASP Conference Series, Vol. 347, 2005
P. L. Shopbell, M. C. Britton, and R. Ebert, eds.

Software Architecture of the Spitzer Space Telescope Uplink/Archive

Joe Chavez, Tatiana Goldina, Annie Hoac, William Roby, and Xiuqin Wu
Spitzer Science Center, California Institute of Technology

Abstract. The Spitzer Science Center (SSC) provides a set of science user tools to support planning and archive access via the Internet. We will present the software architecture and design principles that underlie the Uplink/Archive subsystem of the SSC. Included in the discussion will be a review of the original Uplink architecture as presented in P1-59 ADASS 1999 and the evolutionary changes for the current deployment. The Archive subsystem is based on the same set of core components used in the Uplink subsystem but is based on Web services technology to allow open access to the Archive. Web services technology provides a basis for searching the archive and retrieving data products.

1. Introduction

- SSC Uplink is responsible for the development of software and tools to support the science community with planning and accessing observations with the Spitzer Space Telescope.
- To support the geographically diverse science community the Java 2 Standard Edition platform is the basis for the development of platform independent client software (Spot).
- To support the Spot client software the Java 2 Enterprise Edition is the basis for the development of robust scalable services.
- Also part of the system are C programming language libraries that support for the resource estimation and astronomical visibility services using the Java Native Interface (JNI).
- The software architecture of the SSC Uplink/Archive is presented in the 4+1 views format.

2. Architecture Views

The 4+1 software architecture view divides the architecture into 5 distinct areas of concern (Figure 1). The "+1" view is the Use Case View which contains the set of use cases that define the system behavior. The Logical View represents the logical system structure without regard to implementation specific choices. The Implementation View is the realization of the Logical View in the selected programming languages and tools. The Deployment View shows the physical mapping of components to the hardware nodes in the system. The Process View defines the development processes that are used to construct the system.

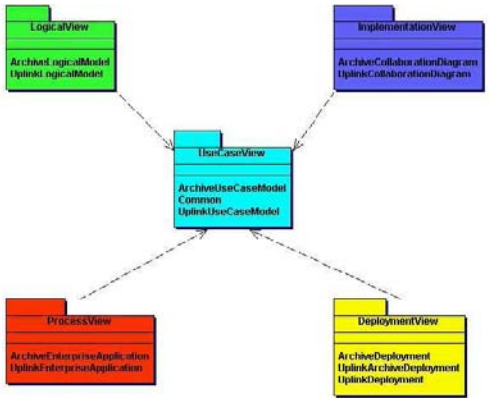

Figure 1. 4+1 Architecture Views

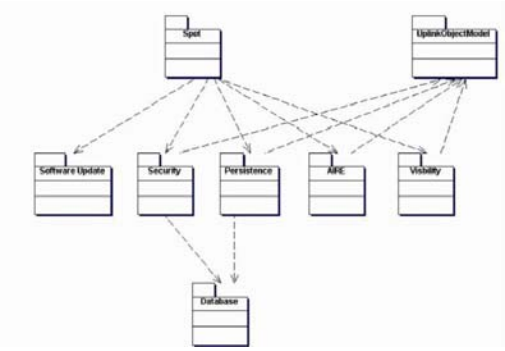

Figure 2. Uplink Component Model

3. Use Case View

The Use Case View contains Use Cases that capture the key functions of the system at a level of detail from which the first iteration of the system architecture can be derived. In the Use Case view there are two main actors: the Astronomer and Operator. The Astronomer represents the class of users of the Spot software. An Operator represents the class of users that operate the Spitzer Science Center.

4. Logical View

This Logical View organizes the component of a system in a vertical stack with increasing order of dependency flowing from top to bottom.

The Uplink component model (Figure 2) consists of the Spot client and the Uplink services. The SoftwareUpdate update component provides a means to update the Spot software at the installed location. This component is split into the client access in Spot and the server management in the J2EE application server. The Security component provides authentication and authorization services to the Spot client. The Persistence component is responsible for managing the persistence of the UplinkComonentModel in the Database. The AIRE com-

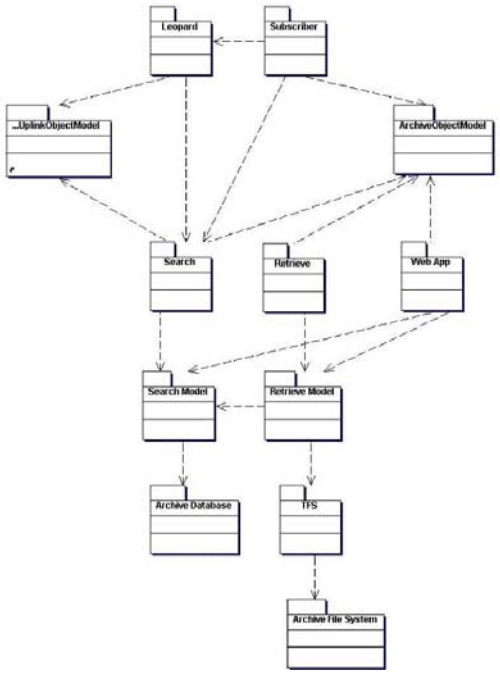

Figure 3. Archive Component Model

ponent is used by Spot to perform resource usage estimation when planning observations for the Spitzer Space Telescope. The Visibility component provides information relating to the ability to view positions at given time with the telescope as well as other estimation services.

The Archive component model (Figure 3) consists of the Leopard and Subscriber clients. The services components are Search and Retrieve. The WebApp component resides on the server and represents the Web browser interface to the Spitzer Archive. The SearchModel provides the data content to the Search service. The RetrieveModel provides the retrieve data to the RetrieveService. The ArchiveDatabase contains the set of Meta-Data for the files contained in the archive. While the SearchModel encapsulates access to the ArchiveDatabase. The RetrieveModel uses the TFS component to access the ArchiveFileSystem via a proxy service provided by the Spitzer Science Data Management group.

5. Implementation View

The Implementation View adds behaviors and interactions to the components defined in the Logical View. In Figure 4, the Uplink component collaborations are shown. This diagram is a representative view of the set of collaborations among the Uplink components. The diagram contains association links between Spot and the various service handler component that provide the implementation. Additional properties of the relationships between the components. The AIREServerHandler component The Archive component collaborations are shown in Figure 5. In this collaboration diagram the primary application flow

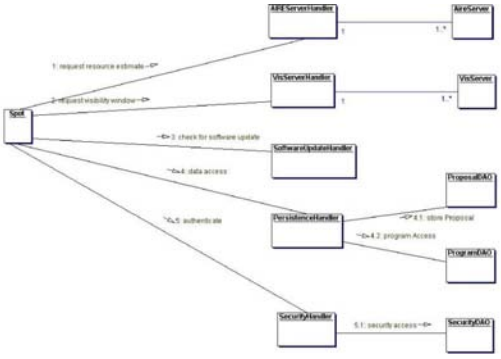

Figure 4. Uplink Collaboration Diagram

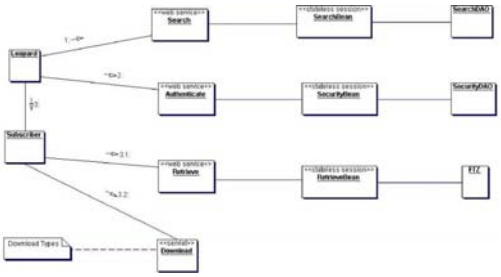

Figure 5. Archive Collaboration Diagram

is depicted.

6. Process View

The Process View encompasses the build and deployment of the system. In the case of the Services Tier the components are packaged into distributable enterprise archive (EAR) files. The EAR files are deployed into a J2EE compliant container. For client software deployment is managed using java archive (JAR) files and native OS installation programs.

7. Deployment View

The deployment diagram shows the combined Uplink/Archive component set as deployed in each tier of the architecture. Each of the components is colored to represent the generation in which the software architecture was designed. The blue colored components represent the original Uplink design circa 1999 and the yellow colored components represent the combined Uplink/Archive design of 2003. The combined Uplink/Archive Deployment model is shown in Figure 6 Some of the key characteristics of this software architecture are:

- Shared Object Model - The Uplink and Archive object model components contain a consistent view of the data that is used across the tiers in the system. In additional to a single view of the data the object model also

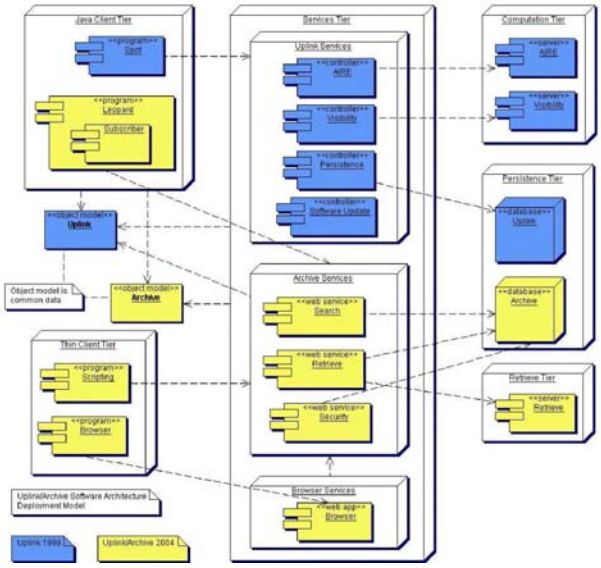

Figure 6. Uplink/Archive Deployment Diagram

provides a set of operations for manipulating the data that helps to maintain data integrity. Internet Protocols - Where appropriate established Internet protocols (HTTP/S) are used for communications between the tiers. In addition, the Archive interface employs Web services via SOAP over HTTP to provide services to any SOAP compatible client.

- Multiple Client Types - The Services Tier exposes multiple interface types to allow many types of clients to access the system services. The types of clients include Java applications, Web browsers and scripting languages (support SOAP).
- Multiple Client Platforms - The Java Client tier allows for the deployment to many client platforms including Unix, Windows, Linux and Mac. The software update service facilitates the update of remote software installations with a minimum of user interaction.
- Distributed Processing - A distributed processing architecture is key to performance and scalability. Where available existing protocols and resource pool management mechanisms are used for inter-process communication. The backend tiers Computation and Retrieve can all scale via server based pool management and clustering. Access to the shared Persistence tier is controlled via JDBC connection pools.